As a permutation, TOF = (7 8). As a mapping of the standard basis it is: ...

Then follows the part to its right in black. The red in the middle notes that to represent multiple wires w out of an input gate x_i, we need to make the ancilla qubit for w have the same value of x_i, so we initialize w to 0 not 1 and make it the target of a CNOT with x_i as source. Basis inputs can be cloned.
I intended to finish by stating the *amplification theorem* for BQP and BPP both: If you have a language \( L \) that belongs to BQP (respectively, BPP) via a predicate \( R(x,y) \) where

\[
\begin{align*}
X \in L & \Rightarrow \Pr_y[R(x,y) > \frac{2}{3}] \\
X \notin L & \Rightarrow \Pr_y[R(x,y) < \frac{2}{3}],
\end{align*}
\]

then for any \( m \) one can define \( R'(x,Y) \) where \( Y \) is a tuple of \( m \) strings to hold if the majority of \( R(x,y_j) \) values hold, \( 1 \leq j \leq m \). This gives

\[
\begin{align*}
X \in L & \Rightarrow \Pr_y[R(x,y) > 1 - \frac{1}{g(m)}] \\
X \notin L & \Rightarrow \Pr_y[R(x,y) < \frac{1}{g(m)}],
\end{align*}
\]

Where the function \( g(m) \) is exponential in \( m \). In particular, with \( m = O(\log n) \), one can get “\( > 1 - 1/3n \)” and “\( \leq 1/3n \)” in the two branches as stated in problem (1) of HW6. You can get “\( > 1 - 1/6n \)” and “\( < 1/6n \)” if you want, or “\( > 1 - 1/n^3 \)” vs. “\( < 1/n^3 \)” or whatever. This is called “amplification by majority vote” and is a powerful technique.